CEWES MSRC/PET TR/99-31

Application of Error Indicators and Local Adaptive Refinement for Elasto-Plastic Impact Calculation (EPIC)

by

Abani K. Patra
Atanas I. Pehlivanov
David Littlefield
Graham F. Carey
J. Tinsley Oden

06h00299

Work funded by the DoD High Performance Computing Modernization Program CEWES Major Shared Resource Center through

Programming Environment and Training (PET)

Supported by Contract Number: DAHC94-96-C0002
Nichols Research Corporation

Application of Error Indicators and Local Adaptive Refinement for Elasto-Plastic Impact Calculation (EPIC)

Abani K. Patra*
Atanas I. Pehlivanov†
David Littlefield‡
Graham F. Carey†
J. Tinsley Oden†

Abstract

We have constructed extensions to the capabilities of the dynamic Lagrangian solid mechanics code EPIC ([10, 11]) to enable basic h-adaptivity on a limited range of problems. The indicators for adaptive refinement include a feature indicator based on relative velocity, a flux jump indicator, hybrid indicators and a stress recovery indicator. We use this capability to test different measures of solution quality and adaptive refinement schemes on a benchmark Taylor anvil test problem.

1 Introduction

Hypervelocity impact simulations present very unique and formidable computational challenges due to the close coupling that occurs between the transport processes and material constitutive behavior. Traditional approaches for solving these types of problems, including Eulerian and Lagrangian methods, have not proven to be entirely successful when applied over a wide range of conditions. For example, Eulerian methods can easily handle large material distortions, but fail to accurately model interface and contact conditions. Lagrangian techniques, on the other hand, can resolve contact problems very precisely, but cannot handle excessive material distortion. We investigate here the use of local adaptive refinement of the finite element meshes involved to improve the quality of the simulation as the mesh deforms. Central to effective

*Department of Mechanical Engineering, SUNY at Buffalo, Buffalo, NY 14260
†Texas Institute for Computational and Applied Mathematics (TICAM), University of Texas at Austin, Austin, TX 78712
‡Institute for Advanced Technology, Austin, TX 78759

use of this technique is the use of effective refinement indicators, adaptation strategies and infrastructure capable of supporting dynamic creation and deletion of elements. We develop and test some representative indicators of solution quality and adaptive refinement strategies.

While the literature on computational impact simulations is vast and there are several comprehensive reviews (see for e.g. Anderson and Bodner [2], Benson[7] for reviews), the use of solution adaptive meshing in these problems is a recent idea. Such a strategy can be seen for example in the recent work of Camacho and Ortiz [12, 13], and Espinosa, Zavatieri and Enmore [9].

2 EPIC code enhancements

The EPIC (Elastic Plastic Impact Calculations) code developed by Johnson et al. [10, 11] uses an explicit Lagrangian finite element formulation. The code has been extensively applied and periodically upgraded over the last two decades. The code is thus now well tested and its capabilities and limitations well known. Access to source code, a simple data structure and robust computational algorithm make it an appropriate choice choice for the preliminary studies of solution adaptive meshing undertaken here.

The equations of motion are integrated directly without forming stiffness matrices. Using lumped masses and initial velocities and displacements, the computation proceeds by computing element strains and stresses, corresponding nodal forces and accelerations followed by an integration of the equations of motion to compute next time-step velocities and displacements. Complex non-linear constitutive models and equations of state are easily incorporated in this cycle. Moving slide lines are used to account for contacting surfaces. Failed elements are removed from the simulation in a process called "erosion".

2.1 Adaptive Methods Infrastructure

The first challenge in using solution based adaptivity to improve finite element codes is the creation of a data structure capable of supporting the creation and deletion of new elements. The EPIC code and most codes designed for simulation of hypervelocity impact include the capabilities of element erosion and re-zoning. These features form the basis for the extensions we provide for automatic element creation and deletion. We use the ability of the code to exclude elements from a simulation, to delete elements that are marked for refinement. We add new elements and nodes to the bottom of the list of existing elements and nodes.While this suffices for the preliminary work described in this report, support of efficient solution-adaptive codes will require a dynamic data-structure with the ability to allocate and deallocate memory as needed. Further, the tree structure implicit in many modern implementations of adaptive refinement will have to be extended to implement unrefinement. This latter capability is significant for wave propagation type problems.

2.2 Element/Node Connectivity and Neighbor Data

Adaptivity and error estimation require frequent reference to information from neighboring elements. The EPIC code was augmented to provide this functionality. Additional data structures were created to store lists of elements associated with each node. The neighbors of an element can be computed efficiently from this information. The refinement process utilizes the neighbor information to propagate the refinement to produce conforming meshes, while the error estimators use it to evaluate quantities like the jump in normal stresses across the inter-element boundary of neighboring elements.

2.3 Grid Transfer Operators

Simple interpolation schemes are used to assign initial values for the various nodal and element quantities to new nodes and elements created. These interpolation routines are part of the re-zoning capability of the EPIC code. Since new elements are created by splitting old elements they inherit all element level state variables from the parent element. Nodes derive their state variables from parent elements or nodes.

3 Error Indicators

Solution adaptive meshing implicitly demands some measure of solution quality. A satisfactory measure is essential for developing a new mesh that will better meet the goals of the analyst. There has been tremendous amount of research on developing such error estimators. Notable among these efforts has been the work of Babuska, Bank, Oden, Zienkiewicz, and their coworkers [5, 4, 3, 17]. The research effort to this date has been primarily directed at elliptic or steady state problems. There are few estimators directed at transient non-linear problems of the type described here. Further, we need estimators that can be easily integrated with the explicit integration type solution methods primarily used in computational impact simulations. In the area of impact simulations, the two categories of indicators that have been looked at have been either primarily geometrical measures like aspect ratios of elements or values of state variables like plastic strain. We consider here some classes of that address both the geometric difficulties encountered and the finite element approximation error.

3.1 RVS indicator

The first refinement indicator we implement is based on physical reasoning. Large and rapid mesh deformation is a primary difficulty in Lagrangian impact simulations and can cause a breakdown in the computations or large errors. We may try to avoid such large mesh deformations by refining the mesh appropriately. If we compute a relative velocity of the nodes of an an element and sum it up for each element then this will serve as a simple measure to identify local mesh deformation. For the typical element JKL (see Fig. 1) define

$$RV_{JK} = \sqrt{(v_{Kx} - v_{Jx})^2 + (v_{Kz} - v_{Jz})^2} \tag{1}$$

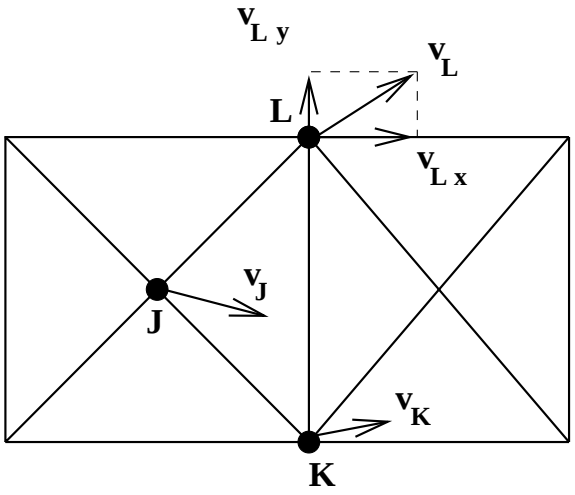

Figure 1: Sample element and velocity vectors of nodes

$$RV_{KL} = \sqrt{(v_{Lx} - v_{Kx})^2 + (v_{Lz} - v_{Kz})^2} \quad (2)$$

$$RV_{LJ} = \sqrt{(v_{Jx} - v_{Lx})^2 + (v_{Jz} - v_{Lz})^2} \quad (3)$$

$$RVS = RV_{JK} + RV_{KL} + RV_{LJ} \quad (4)$$

The quantity RVS can be used as an indicator of evolving mesh deformations. However, it is a purely geometric indicator and has no systematic connection to the error in the underlying computations. This indicator is also similar in spirit to indicators chosen by Ortiz et al. [12] and Batra et al. [6].

3.2 Flux-jump Based Indicator

In any C^0 finite element simulation the discretization error will cause the inter-element fluxes to be discontinuous. Conversely one may use this inter-element flux discontinuity as an indicator of the error in the simulation. A major component of the error residual [3] does indeed consist of appropriately distributed jumps in the normal traction at each inter-element boundary.

We compute for each element and it's three neighbors the stresses normal to the boundary and designate them σ_L and σ_R respectively (see Fig. 2). We then compute the jump $[\sigma_n]$ in these stresses and its norm $|||[\sigma_n]|||_{\partial\Omega_e}$ according to

$$[\sigma_n] = \sigma_L - \sigma_R \quad (5)$$

$$|||[\sigma_n]|||_{\partial\Omega_e} = \int_{\partial\Omega_e} [|\sigma_n] ds \quad (6)$$

This is then distributed equally to each element sharing the edge. While this indicator may correlate well with the finite element approximation error it does not predict the onset of large mesh deformation that corrupt the Lagrangian simulations of impact phenomena.

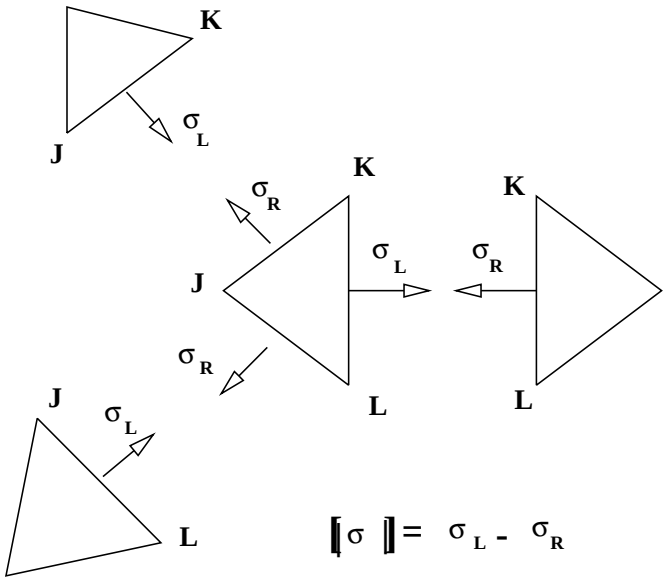

Figure 2: Quantities involved in flux jump computation

3.3 Hybrid Indicators

In these indicators we attempt to combine the attributes of the two previous indicators using appropriate scaled and weighted combinations to derive new hybrid estimators.

1. **Additive Combination** In the first combination we obtain the refinement indicator for element Ω_e using a linear combination of the two previously defined indicators RVS_e and $|||[\sigma_n]|||_{\partial\Omega_e}$,

$$H_e = \alpha \frac{RVS_e}{\max_e\{RVS_e\}} + \beta \frac{|||[\sigma_n]|||_{\partial\Omega_e}}{\max_e\{|||[\sigma_n]|||_{\partial\Omega_e}\}} \tag{7}$$

   where α, β are empirically chosen scalars and the maximum values over the entire grid are used for normalization. The simplest choice of the scalars is $\alpha = \beta = 1$.

2. **Multiplicative Combination:** In the second combination the refinement indicator is obtained by a multiplicative combination of the RVS and flux jump indicators. We can write this indicator as

$$E_e = \frac{RVS_e}{\max_e\{RVS_e\}} \cdot \frac{|||[\sigma_n]|||_{\partial\Omega_e}}{\max_e\{|||[\sigma_n]|||_{\partial\Omega_e}\}} \tag{8}$$

   We note here that this indicator is similar to the family of error estimators proposed by Ranacher et al. [15, 16] for steady state problems. This indicator is described as

$$||e||_{\Omega_e} = \omega_e \cdot \rho_e \tag{9}$$

   where ρ_e is an element residual computed using a locally enhanced approximation and inter element flux jumps (see, for example, Akin [1]) and ω_e is a suitable

weight function – usually a local Greens function where pointwise error control is a desired goal. Our multiplicative indicator can be interpreted in terms of this estimator since the flux jump term is the major component of the residual and the RVS quantity is viewed as the weight function of choice for this application.

3.4 ZZ-type indicator

This section outlines the implementation of a recovery indicator of Zienkewicz-Zhu type. Let u_h be a continuous piecewise linear function that is a finite element approximation of u on the domain of interest. Our goal is to calculate a continuous piecewise linear vector function that will be an approximation to the gradient of u. More, specifically, a recovery operator G is defined such that

$$G[u_h] \approx \mathbf{grad}\ u$$

Let x_m be a node of some element in the finite element triangulation T_h. First, we have to identify a patch Π_m of elements having a vertex at x_m, i.e.

$$\Pi_m = \{K \in T_h \ :\ x_m \text{ is a vertex of } K\}$$

Define z_K as the barycenter of the element K. Let g be a vector function with linear components defined on the patch Π_m, i.e.

$$\boldsymbol{g}(x) = \begin{pmatrix} ax + by + c \\ Ax + By + C \end{pmatrix}$$

Now we find the coefficients of g by calculating a least-squares fit to the gradient of u_h sampled at the baricenters of all elements in the patch Π_m

$$\boldsymbol{g}(x) = \min_{a,b,c,A,B,C} \sum_{K \in \Pi_h} \left(\boldsymbol{g}(z_K) - \mathbf{grad}\ u(z_K)\right)^2$$

Then the value of $G[u_h]$ at the node x_m is set to be

$$G[u_h](x_m) = \boldsymbol{g}(x_m)$$

Applying the above procedure to all nodes in the triangulation completes the construction of the recovery operator $G[u_h]$.

When the node x_m lies on the boundary of the domain, we may not have enough neighbor elements, i.e. the least squares fit may not be unique. In this case we enlarge the patch by adding all elements that are neighbors to the neighbors.

First, we apply the recovery operator to obtain smooth stress fields. Then we combine the components of the difference between the recovered stresses and the corresponding finite element approximations using a von Mises type approach and denote that quantity by $(\sigma^{recovered} - \sigma^h)_{VM}$. A particular element is refined if

$$(\sigma^{recovered} - \sigma^h)_{VM} > 0.5 \max_{K \in T_h} \left\{ (\sigma^{recovered} - \sigma^h)_{VM} \Big|_K \right\}$$

where $(\sigma)_{VM}$ is defined as

$$\sigma_{VM} = \sqrt{0.5\left[(\sigma_x - \sigma_y)^2 + (\sigma_x - \sigma_z)^2 + (\sigma_y - \sigma_z)^2\right] + 3(\tau_{xy}^2 + \tau_{xz}^2 + \tau_{yz}^2)}$$

Another quantity that may be used in the refinement process is the "energy" norm of the error on each element defined as

$$\|e\|_K^2 = \int_K (\sigma^{recovered} - \sigma^h)(\epsilon^{recovered} - \epsilon^h)\ dx$$

Again, we refine an element K if

$$\|e\|_K^2 > 0.5 \max_{K \in T_h} \left\{ \|e\|_K^2 \right\}$$

4 Adaptive Strategy

The goal of an adaptive strategy is to develop a new improved mesh based on error indicator or other measure of solution quality. In the case of the hypervelocity impact simulations where excessive deformations cause simulations to fail, we will attempt to develop adaptive strategies that predict such behavior and create new elements and/or reposition nodes and elements to avoid these problems. Additionally we would like to control discretization error at a satisfactory level. However, at this time this is a secondary goal.

4.1 Types of Refinement Allowed

We allow three basic types of refinement on the axisymmetric triangles that are used to mesh our test problem. Since, the EPIC code lacks the ability to support hanging/constrained nodes we have to produce a conforming mesh. This will lead to refinements on elements adjoining the target element and beyond in a recursive fashion.

If the element to be refined and it's neighbor both share their longest edge, it is a simple matter to refine them both. We designate this type of refinement Type I(see Fig. 3). If the element and its neighbor do not share the longest edge we allow the refinement to propagate. We split the longest edge of the neighbor and enforce a split on its neighbor. We then split the neighbor in three and original element in two. We designate this refinement Type II (see Fig. 3). This simple restriction to only two levels is indeed somewhat ad-hoc and enforced to keep the program complexity manageable and reduces the needed data-structure support.

In the third type of refinement we split an element in 3 by joining its centroid to each node. However, this type of refinement is unsuitable if the element being split has a poor aspect ratio since the child elements have aspect ratios that are worse than the parent element. We use this refinement only if the parent meets a stringent aspect ratio criteria.

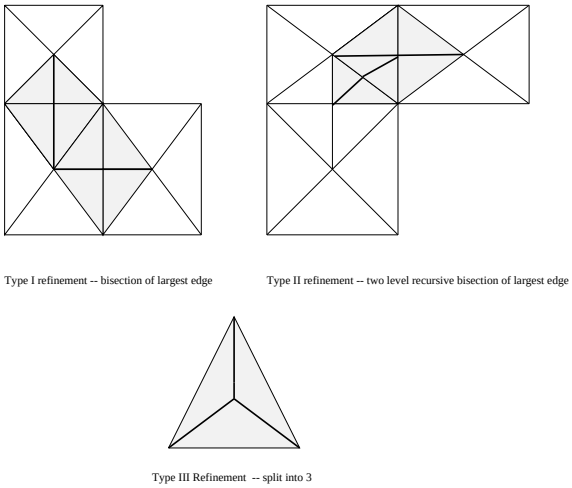

Figure 3: Mesh Patches and types of refinement allowed

4.2 Types of Unrefinement Allowed

Removing elements to improve the geometry of remaining elements is a much more complex operation. The classical way removing elements in a zone and replacing them with a new local remesh called rezoning has been a part of EPIC and most such codes. We experiment here with a a slightly different approach. We allow a few limited types of local unrefinement where only elements and immediate neighbors are affected (see Fig. 4). Unrefinement is triggered by the same element based measures as are used for refinement. A more systematic unrefinement scheme where the size of patch to unrefine is determined adaptively will be undertaken in future work.

4.3 Strategy

In the first strategy we use a simple fixed fraction refinement strategy. Element refinement indicators in the top 20 or 30 % are refined after each 25% of total simulation time. A better approach would be to trigger refinement after a certain value of the error indicator is achieved. However, no reliable error measures are readily available. We are currently developing more targeted refinement strategies [14] where the error is spatially equidistributed and refinement is targeted to obtain a desired error level.

5 Results

We implement the above refinement indicators and adaptive strategies for a Taylor impact simulation. This is a much studied classical problem in impact calculations. In our first numerical test we shall use the RVS indicator and the above described refinement strategy. We show a sequence of computations over a time period of 0.1E-04 seconds for a cylindrical rod of iron striking a rigid surface. Figure 5 shows the initial mesh and geometry. Figure 6 shows the deformed geometry and the values of

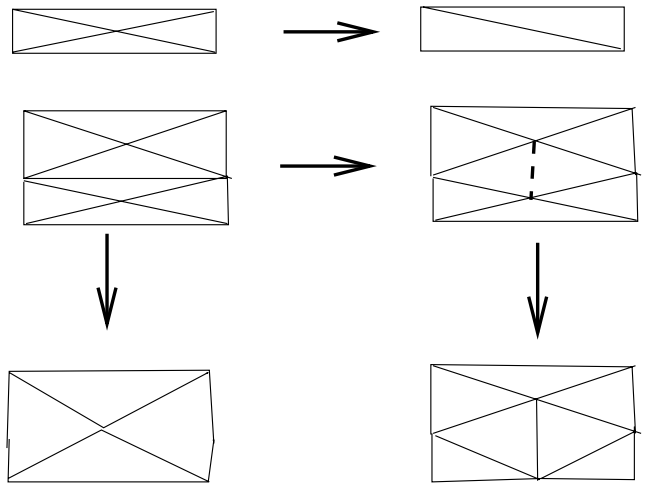

Figure 4: Mesh Patches and types of unrefinement allowed

the refinement indicator and plastic strain distribution and the corresponding adapted mesh at 0.1E-04 seconds. The strategy adds 32 nodes and 155 new elements. The refinements are clearly focussed in the region where maximum element distortion is experienced. However, the plots of plastic strain indicate that the highly strained region in the bottom outer zone of the cylinder does not have much refinement. In the next test we use the flux jump indicator and the fixed fraction refinement strategy with top 30% of the elements being refined. The computations are again carried out over the same time period of 0.1E-04 seconds. Figure 7 show the results of this simulation. Adapted meshes, contours of $|||[|\sigma_n|]|||_{\partial\Omega_e}$ and plastic strain at at 0.03E-04,0.051E-04, 0.075E-04 and 0.1E-04 seconds are shown. In the third set of tests we use the additive hybrid indicator H_e with $\alpha = \beta = 1$ indicators with a fixed fraction refinement strategy with refinement triggered whenever $H_e > 1.$. The computations are again carried out over the same time period of 0.1E-04 seconds. Figure 8 show the results of this simulation. Adapted meshes, contours of $|||[|\sigma_n|]|||_{\partial\Omega_e}$ and plastic strain at 0.03E-04,0.051E-04, 0.075E-04 and 0.1E-04 seconds are shown. This indicator and strategy add 55 nodes and 201 elements. The refinements are now distributed more evenly. In the fourth set of tests we use the hybrid indicator E_e indicators with a fixed fraction refinement strategy with refinement triggered whenever $E_e > 0.3$. The computations are again carried out over the same time period of 0.1E-04 seconds. Figure 9 show the results of this simulation. Adapted meshes, contours of $|||[|\sigma_n|]|||_{\partial\Omega_e}$ and plastic strain at 0.03E-04,0.051E-04, 0.075E-04 and 0.1E-04 seconds are shown. In the next set of results we investigate the performance of the ZZ type error indicator. Fig. 10 shows the error indicators obtained.

Finally, we combine refinement and unrefinement and look at its effect on the aspect ratios of elements. The average aspect ratios are now dramatically improved as seen on Fig. 11. However, the minimum aspect ratio has not improved much.

6 Conclusions and Future Work

In this report we have examined a few basic error indicators and adaptive strategies for Lagrangian impact calculations. Preliminary results are encouraging and the approach leads to improved meshes for the simulations. However, much remains to be done to use these indicators and strategies for realistic impact simulations. Principal among the tasks that need to be accomplished are creation of a dynamic data structure capable of supporting creation and deletion of elements, implementation of mesh smoothing and local remeshing algorithms to augment element refinement and coarsening, development of more reliable error estimators as opposed to the simple error indicators considered here, study of the interaction of adaptivity and different contact algorithms, and integration of the mesh refinement, smoothing and remeshing and different types of error indicators with an appropriate choice of technique based on values of different indicators.

Acknowledgements

This work has been partially supported by the DoD High Performance Computing Modernization Program, CEWES Major Shared Resource Center, through Programming Environment and Training (PET). We would also like to express our appreciation to Richard Weed for his helpful comments.

Views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy or decision unless so designated by other official documentation.

References

[1] J. E. Akin, Finite Elements for Analysis and Design, Academic Press, New York, 1994.

[2] C. E. Anderson Jr. and S. R. Bodner, "Ballistic Impact: The Status of Analytical and Numerical modelling," *Int. J. Impact Engg.* vol 16, 1988 pp. 9-35.

[3] M. Ainsworth , J. T. Oden,"*A Posteriori Error* Estimators, *Computational Mechanics Advances, Computer Methods in Appl. Mech. and Engg. ,* to appear.

[4] R. E. Bank and R. K. Smith, "*A Posteriori* Error Estimates Based on Hierarchical Bases", Institute for Mathematics and its Applications, Summer Program, July 1993.

[5] I. Babuska and W.C. Rheinboldt, "Error Estimates for adaptive finite element computations", International Journal for Numerical Methods in Engineering, Vol. 12, 1978, pp.1597-1615.

[6] R. C. Batra and K. I. Ko, "An adaptive mesh refinement technique for the analysis of shear bands in plain strain compression of a thermoviscoplastic solid" *Comp. Mech.* Vol 10, 1992, pp. 369-379.

[7] D. J. Benson, "Computational Methods in Lagrangian and Eulerian hydrocodes", *Comput. Methods.Appl. Mech. and Engg.* vol. 99 1992, pp. 235-394.

[8] L. Demkowicz, J. T. Oden, W. Rachowicz, O. Hardy, "Towards a Universal *hp* Adaptive Finite Element Strategy, Part 1 Constrained Approximation and Data Structure ", *Comput. Methods.Appl. Mech. and Engg.*, 77(1989), pp 79-112.

[9] H. D. Espinosa, Pablo D. Zavattieri and G. L. Emore, "Adaptive FEM computation of geometric and material non-linearities with application to brittle failure", preprint to appear in *Mechanics of Materials*.

[10] G. R. Johnson and R. A. Stryk, "Eroding interface and improved tetrahedral element algorithms for high-velocity impact computations in three dimensions", *Int. J. Impact Engg.* vol. 5 1987, pp. 411-421.

[11] G. R. Johnson and R. A. Stryk, "User Instructions for the EPIC-3 code" Air Force Armanent Laboratory.

[12] G T. Camacho and M. Ortiz, "Adaptive Lagrangian Modelling of ballistic penetration of Metallic Targets", preprint, (to appear in *Comput. Methods.Appl. Mech. and Engg.*)

[13] M. Ortiz and I.V. Quigley, "Adaptive mesh refinement in strain localization problems", *Comp. Meth. App. Mech.* vol 90 , 1991 pp.781-804.

[14] J. T. Oden and Abani Patra, "A Parallel Adaptive Strategy for *hp* finite element computations", *Comput. Methods. Appl. Mech. and Engg.*, vol. 121, 1995.

[15] R. Rannacher and F. Theo Suttmeier, "A feedback approach to Error Control in Finite Element Methods: Application to Linear Elasticity", Preprint 96-42, IWR, Heidelberg, September, 1996.

[16] R. Becker, and R. Ranacher, " A feedback approach to Error Control in Finite Element Methods:basic analysis and examples", East-West J. Numerical Mathematics, Vol.4 , No. 4, 00.237-264, 1996.

[17] O. C. Zienkewicz and J. G. Zhu, "A simple error estimator and adaptive strategy for practical engineering analysis", *International Journal for Numerical Methods in Engineering*, vol. 24, 1987, pp. 337-357.

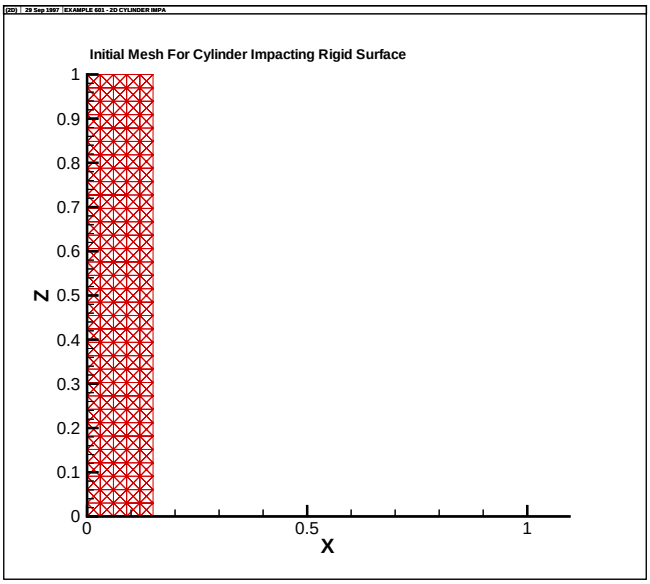

Figure 5: Initial Mesh for cylindrical rod impacting rigid surface

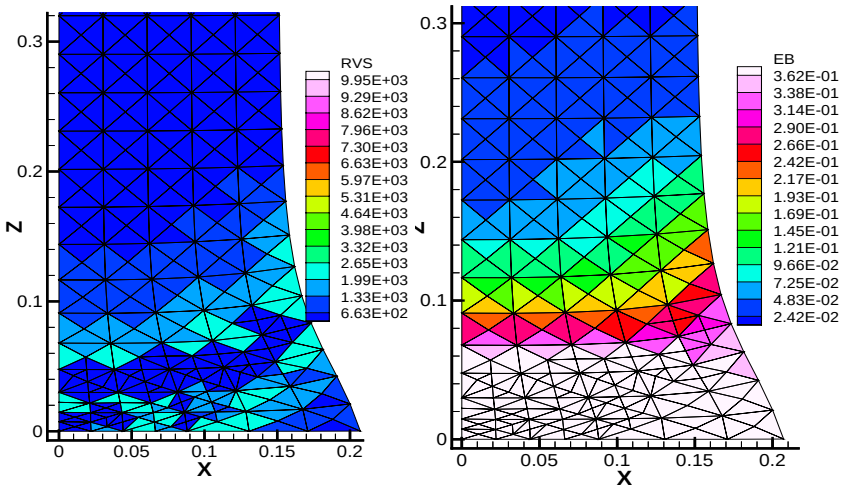

Figure 6: Mesh and contours of RVS and plastic strain for cylindrical rod impacting rigid surface at time = 0.1E-04 after multiple adaptation cycles using RVS indicator.

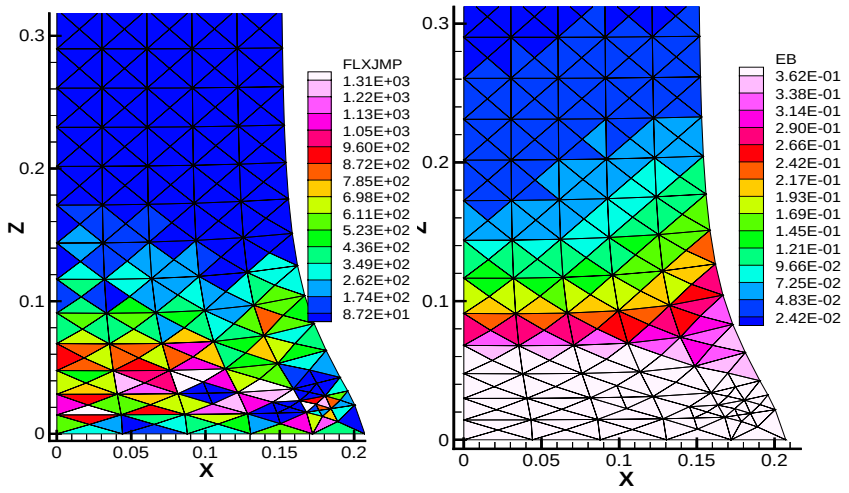

Figure 7: Mesh and contours of flux jump and plastic strain at time = 0.1E-04 for cylindrical rod impacting rigid surface after multiple adaptation cycles using flux jump indicator.

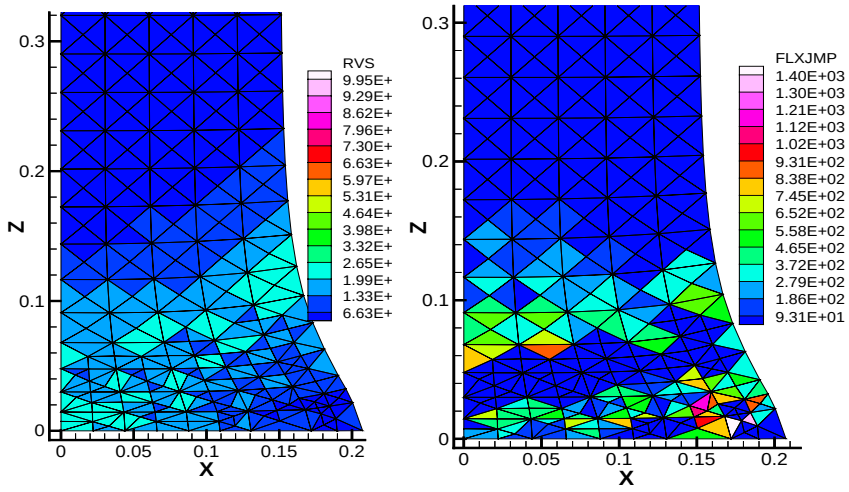

Figure 8: Mesh and contours of plastic strain, RVS and flux jump at time = 0.1E-04 for cylindrical rod impacting rigid surface after multiple adaptation cycles using additive hybrid indicator.

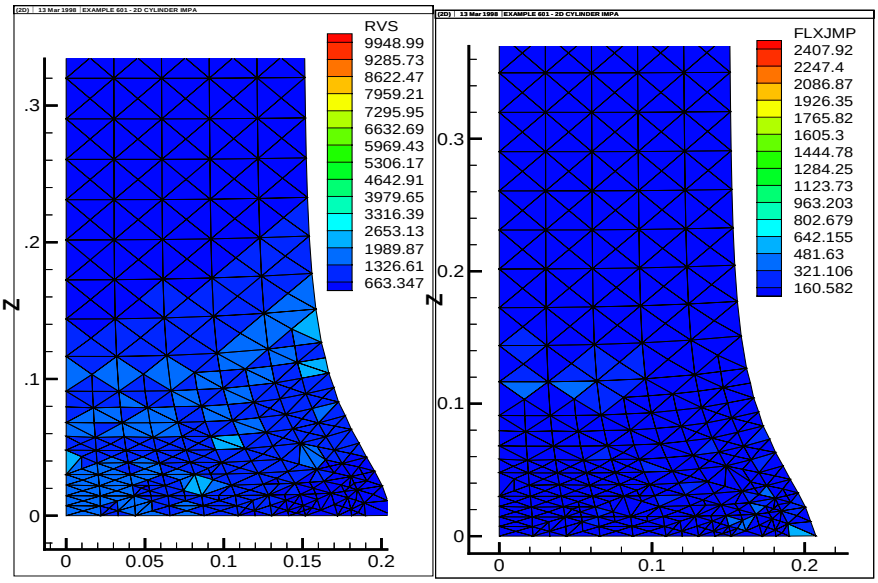

Figure 9: Mesh and contours of error indicator, RVS and flux jump at time = 0.1E-04 for cylindrical rod impacting rigid surface after multiple adaptation cycles using multiplicative hybrid indicator.

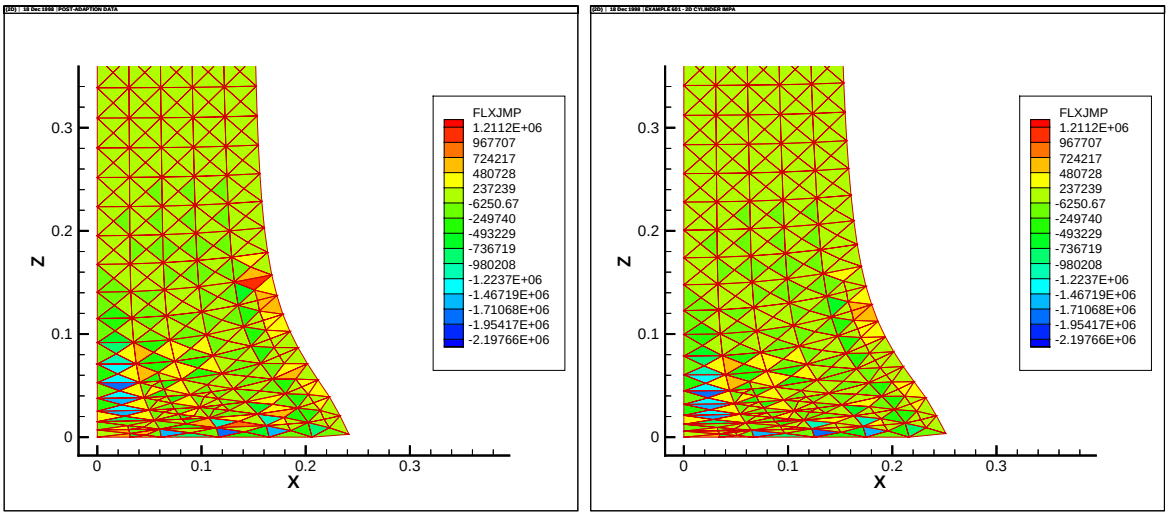

Figure 10: Contours of adapted mesh and energy associated with ZZ type error at time 0,225030E-04 and .300046E-04 sec.

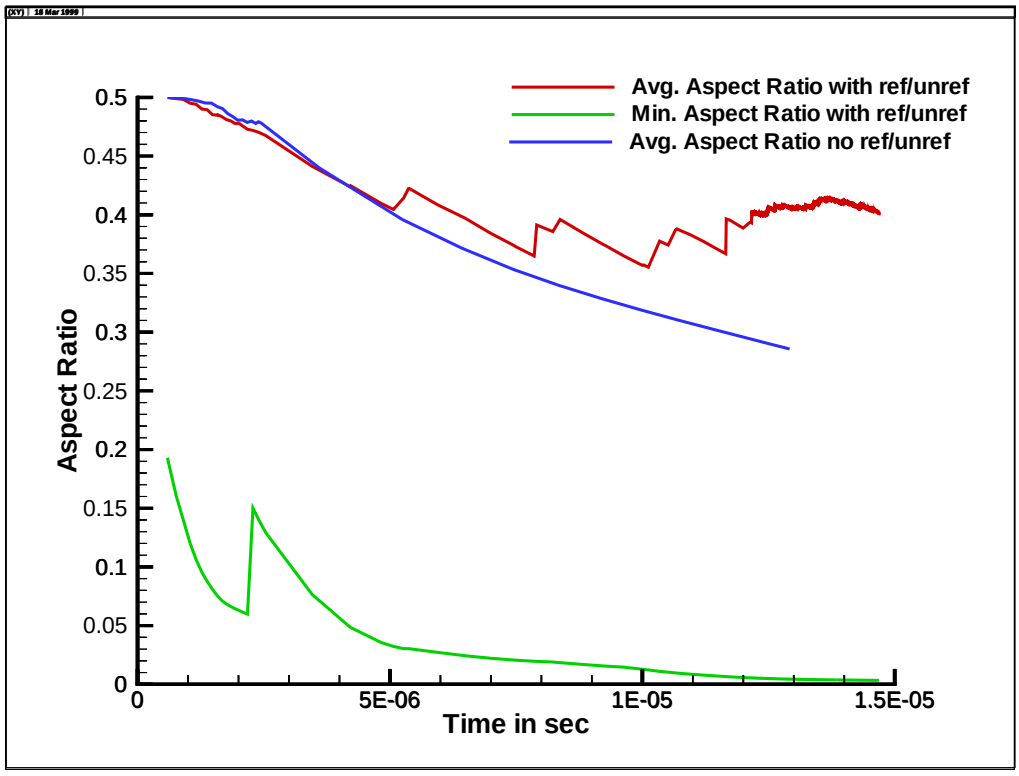

Figure 11: Aspect Ratios as the mesh is refined and unrefined